

TP LINUX GESTION DES UTILISATEURS ET DES GROUPES

1- Comptes d'utilisateurs et de groupes

Quand un utilisateur se connecte, il s'identifie en fournissant un nom de connexion (*login*) et un mot de passe (*password*). Si la connexion réussit, l'utilisateur a normalement activé un **shell** (*interpréteur de commandes*) et se trouve positionné dans son répertoire de travail.

L'ensemble des informations qui caractérisent les paramètres fondamentaux de la connexion d'un utilisateur, y compris de l'utilisateur root, sont définies dans le fichier `/etc/passwd`. Le fichier `/etc/group` définit les groupes existants sur le système.

1.1. Structure du fichier `/etc/passwd`

Le fichier `/etc/passwd` est un fichier de type texte dont chaque ligne définit un compte utilisateur. La ligne est composée de champs. Le séparateur de champs est le caractère ":".

```
# more /etc/passwd
root:x:0:0:root:/root:/bin/bash
svidal:x:1009:1011:Sophie Vidal:/home/etudiant/bts/svidal:/bin/bash
```

Nom de connexion	Le nom de connexion de l'utilisateur.
Mot de passe	Le mot de passe est présent dans le fichier, mais de manière cryptée. Dans les systèmes sécurisés, le champ <i>mot de passe</i> existe toujours, mais il contient le caractère "x". Le mot de passe crypté est en fait déporté dans un fichier accessible au seul administrateur (<code>/etc/shadow</code>). Les caractères "!" dans le champs <i>mot de passe</i> indiquent que le mot de passe n'est pas défini, si les mots de passe sont visibles.
UID	L'administrateur attribue un numéro à chaque utilisateur. Ce numéro, l'UID, est mémorisé dans les descripteurs de fichiers pour en identifier le propriétaire. L'UID de root est 0.
GID	L'administrateur identifie le groupe de connexion d'un utilisateur grâce au champ GID (<i>Group Identification</i>). Le fichier <code>/etc/group</code> associe un nom de groupe à ce GID et définit les groupes supplémentaires de l'utilisateur.
Commentaire	La zone est utilisée librement par l'administrateur pour commenter le compte.
Répertoire de connexion	Ce champ détermine le répertoire de connexion de l'utilisateur (<i>répertoire de travail</i>), conventionnellement <code>/home/wwallace</code> pour un utilisateur de nom de connexion <code>wwallace</code> . Ce répertoire contient les fichiers de configuration (<code>.bash_profile</code>) de l'utilisateur.
Commande de connexion	Ce champ précise le chemin d'accès absolu de la commande à exécuter lors de la connexion. En général, c'est un <i>shell</i> .

1.2. Structure du fichier `/etc/group`

```
# more /etc/group
invite:x:1009:
btsga:x:1011:
```

Nom du groupe	Le nom du groupe !!!
Mot de passe	Le mot de passe est présent dans le fichier de manière crypté. Il est demandé à un utilisateur qui veut rejoindre le groupe. Rarement utilisé en pratique.
Liste des utilisateurs	La liste des utilisateurs qui ont ce groupe comme groupe secondaire.

1.3. Commandes de gestion des utilisateurs

Nom de la commande	Action
Commandes d'administration	
<code>useradd [-c comment] [-d home_dir] [-e expire_date] [-f inactive_time] [-g initial_group] [-G group[,...]] [-m] [-k skeleton_dir] [-M] [-p passwd] [-s shell] [-u uid] login</code>	Permet de créer des comptes utilisateur et de définir les paramètres qui seront utilisés lors de la création du compte.
<code>usermod [-c comment] [-d home_dir] [-e expire_date] [-f inactive_time] [-g initial_group] [-G group[,...]] [-m] [-k skeleton_dir] [-M] [-p passwd] [-s shell] [-u uid] login</code>	Permet de modifier un compte utilisateur.
<code>userdel login</code>	Suppression un compte utilisateur.
<code>groupadd [-g gid] [-o] [-r] [-f] group</code>	Permet de créer un groupe.
<code>groupmod [-g gid] [-o] [-r] [-f] group</code>	Permet de modifier un groupe.
<code>groupdel group</code>	Suppression d'un groupe d'utilisateur.
<code>pwck</code>	Vérifie l'intégrité des fichiers passwd ou shadow. Chaque entrée est vérifiée pour voir si chaque champ contient des informations de format correct. L'utilisateur se verra demander s'il veut effacer chaque entrée incorrecte.
<code>grpck</code>	Idem pwck pour le fichier /etc/group.
Commandes d'information et de gestion des utilisateurs	
<code>finger</code>	Permet d'obtenir des informations sur un utilisateur.
<code>users</code>	Affiche le nom des utilisateurs actuellement connectés.
<code>groups</code>	Affiche les groupes auxquels appartient un utilisateur.
<code>id</code>	Affiche l'UID et le GID d'un utilisateur.
<code>su [login]</code>	Permet d'exécuter un shell avec un UID/GID différent. Si on ne précise pas de nom d'utilisateur, l'utilisateur essaie d'exécuter un <i>shell</i> en tant que <i>root</i> (<i>très pratique pour exécuter temporairement une commande en tant que root</i>).
<code>passwd</code>	Permet de créer, modifier un mot de passe ou de verrouiller/déverrouiller un compte.

Options de useradd	Description	Options de groupadd	Description
<code>-c</code>	Commentaire	<code>-g</code>	GID. GID>500 pour les groupes utilisateurs
<code>-d</code>	Répertoire de connexion	<code>-o</code>	Autorise plusieurs groupes à avoir le même GID.
<code>-e</code>	Date d'expiration du compte	<code>-r</code>	Création d'un groupe système (GID<400)
<code>-f</code>	Le nombre de jours au bout duquel le compte est inutilisable, après la date d'expiration		
<code>-g</code>	Le groupe de l'utilisateur. Par défaut, Linux crée un compte de même nom que l'utilisateur		
<code>-G</code>	Les groupes supplémentaires		
<code>-m</code>	Obligation de créer le répertoire de l'utilisateur		
<code>-k</code>	Les fichiers contenu dans <code>skeleton_dir</code> seront recopiés dans le répertoire personnel. Par défaut, il s'agit de <code>/etc/skel</code> .		
<code>-M</code>	Le répertoire de l'utilisateur n'est pas créé.		
<code>-p</code>	Le mot de passe		
<code>-s</code>	Le <i>shell</i> de l'utilisateur, par défaut <code>bash</code> .		
<code>-u</code>	UID de l'utilisateur		

2- Droits

Sous Linux, n'importe quel fichier ou répertoire appartient à un utilisateur, qui lui-même appartient à un groupe. Les droits sont répartis de la manière suivante :

Exemple : `rwxr--r-- 1 wwallace jury 510 sep 17 11:20 notes`

2.1. Commande de gestion des droits

Nom de la commande	Action
<code>ls -l nom_fichier</code>	Connaître les droits d'un fichier/répertoire
Deux méthodes : <code>chmod ugoa {+ -=} rwx fichier</code> <i>u : user ; g : groupe ; o : other ; a : tout le monde</i> <code>chmod mode fichier</code>	Changer les droits d'un fichier/répertoire
<code>chown nom_nouveau_propriétaire fichier</code>	Changer le propriétaire d'un fichier/répertoire
<code>chgrp nom_nouveau_groupe fichier</code>	Changer le groupe d'un fichier/répertoire

2.2. Exemples de modifications de droits d'un fichier/répertoire

Syntaxe : `chmod ugoa {+|-=} rwx nom_fichier`

Exemple : `chmod a-rwx,u+rwx nom_fichier`

Retire tous les droits sur le fichier sauf pour l'utilisateur qui aura le droit de lire, écrire et exécuter.

Syntaxe : `chmod mode nom_fichier`

Exemple : `chmod 700 nom_fichier`

mode représente, en base 8, les droits du fichier. Trois termes représentent les droits. Chaque terme est regroupé en puissance de 2 ($r = 4, w = 2, x = 1, - = 0$).

Donne tous les droits au propriétaire ($7 = 4 + 2 + 1$ donc `rw``x`) et aucun droit au groupe ni aux autres.

3- Première application (tutoriel)

1. Créer un utilisateur toto "toto", **adduser toto**
Saisissez puis confirmez le mot de passe. La création de l'utilisateur entraîne la création de son dossier personnel : /home/toto
2. Regarder l'UID associé à l'utilisateur "toto", **grep toto /etc/passwd**
Le fichier **/etc/passwd** contient tout ce qui concerne la gestion et l'authentification des utilisateurs. Chaque ligne est propre à un utilisateur et comprend 7 champs séparés par le symbole [:].
3. Regarder le GID associé à l'utilisateur "toto", **grep toto /etc/group**
4. A l'aide de la commande **id**, vérifier votre identité et le(s) groupe(s) au(x)quel(s) vous appartenez.
5. Basculer dans le profil toto, **su – toto**
6. Créer un fichier "toto.txt", **touch toto.txt**
7. Regarder les droits affectés à "toto.txt", **ls -l toto.txt**
8. Quitter le profil toto, **logout**
9. Changer le mot de passe (toto ⇒ passtoto) **Passwd toto**
10. Afficher les groupes : **cat /etc/group**
11. Ajouter un groupe : **Addgroup ami**
12. Modifions à présent le groupe de notre utilisateur toto (Il fait partie du groupe ami)
pour changer toto de groupe nous utiliserons la commande **usermod**: **usermod -g ami toto**
toto ne fait donc plus partie du groupe toto mais du groupe ami.
avec l'argument -G en majuscule, il est possible d'affecter à toto plusieurs groupes :
usermod -G toto,ami toto
13. Afficher les groupes d'un utilisateur : **groups toto**
Le fichier **/etc/group** contient tout ce qui concerne les groupes. Chaque ligne est propre à un groupe et comprend 4 champs séparés par le symbole [:].
Voici la commande pour rétablir le groupe d'origine :


```
usermod -g mon_groupe_principal mon_utilisateur (usermod -g toto toto)
adduser mon_utilisateur groupe_secondaire
```


usermod permet donc de modifier la configuration d'un utilisateur.
usermod pouvait modifier l'appartenance à un group utilisateur.
Mais permet aussi de modifier le nom d'un utilisateur ou UID.
Ajouter des informations à un utilisateur Linux : **usermod -c "C'est un super utilisateur" toto**
14. Supprimer un groupe : **delgroup ami**
15. Supprimer l'utilisateur toto, **userdel toto**
userdel ne supprime pas son home /home/toto, ainsi il faudra le faire manuellement. Notez bien que n'importe quel utilisateur ajouté par la suite avec le même UID/GID que le précédent propriétaire aura alors accès à ce dossier, si vous n'avez pas pris les précautions nécessaires.
Par supprimer le dossier toto il faut compléter avec la commande suivante : **sudo rm -R /home/toto**
Suppression d'un user avec son dossier perso : **userdel -r toto** l'argument -r permet la suppression du dossier perso
16. Regarder les droits affectés à "toto.txt" **ls -l /home/toto/toto.txt**

TP – Linux : gestion des Utilisateurs et des Groupes

Si on lance la commande **ls -l** on voit quelque chose comme : **d rxw rxw rxw**
Le premier caractère représente le type de fichier:

- : un fichier	d : directory / dossier	l : link / lien	s : socket p : fifo	c : device (mode caractère)	b : device (mode block)
----------------	--------------------------------	------------------------	-----------------------------------	------------------------------------	--------------------------------

viennent ensuite 3 triplets

1er triplet rxw : droits du propriétaire	2em triplet rxw : droits du groupe	3em triplet rxw : droits des autres utilisateurs
--	------------------------------------	--

leur signification est la suivante:

r (read) : l'élément peut être lu	w (write) : l'élément peut être modifié/supprimé	x (execute) :	- (tiret) : l'élément n'a pas de droit
--	---	----------------------	--

- 1) si l'élément est un programme ou un script on peut l'exécuter.
- 2) si c'est un dossier **x** signifie que l'on peut voir ses sous répertoires qui, eux-mêmes ont un droit en lecture (**r**)

Il est possible d'utiliser **chmod** différemment, il faut alors savoir que :

u = user	g = group	o = other
-----------------	------------------	------------------

et que

+ ajoute le droit	- retire le droit	= affecte le droit
--------------------------	--------------------------	---------------------------

On quitte le shell de toto et on supprime cet utilisateur. Puis on regarde les droits associés à ses anciens fichiers. On constate en outre que ses fichiers n'ont pas été supprimés. Effectivement, ceci est le rôle de la commande "rm" et non userdel. En effet, supprimer un utilisateur ne signifie pas que l'on veut nécessairement supprimer ses fichiers.

On s'aperçoit que comme toto (ni le groupe toto) n'est référencé dans /etc/passwd (et dans /etc/group), la commande ls ne parvient plus à retrouver à quoi correspondent l'UID et le GID 1001.

Voici les conséquences immédiates : Certains utilisateurs peuvent ne plus pouvoir accéder à ces fichiers.

Par la suite, cela peut être catastrophique. Pour voir plus clair, créons un utilisateur titi, examinons son UID et les droits associés à toto.txt :

Comme l'utilisateur toto vient d'être supprimé, titi a récupéré l'UID que toto utilisait. Examinons les droits associés à toto.txt :

Comme titi s'est accaparé l'UID de toto, le fichier toto.txt lui appartient ! Pour éviter cela, il aurait fallu au moment de créer titi lui attribuer un autre UID, ou réattribuer les fichiers de /home/toto à un autre utilisateur (par exemple root).

- | | |
|---|---------------------------|
| 17. Ajouter un utilisateur titi : | adduser titi |
| 18. Vérifier le UID et GID de titi : | grep titi /etc/passwd |
| 19. Afficher les groupes : | grep titi /etc/group |
| 20. Regarder les droits affectés à "toto.txt" : | ls -l /home/toto/toto.txt |
| 21. Supprimer les groupes et les utilisateurs créés pendant cette activité avec les commandes appropriées | |

userdel -r titi
sudo rm -R /home/toto

4- Deuxième application

Question 1 : Créer les groupes btsga et btsgb.

Question 2 : Vérifier que les groupes créés précédemment figurent dans le fichier des groupes.

Question 3 : Créer l'utilisateur etudiant1 BTS, dont le *login* est e1BTS, qui appartient au groupe btsga.

Question 4 : Vérifier que l'utilisateur e1BTS a bien été créé.

Question 5 : Quel est le répertoire de base de l'utilisateur e1BTS ?

Question 6 : Quel est l'interpréteur de commandes de l'utilisateur e1BTS ?

Question 7 : Vérifier que le répertoire personnel de l'étudiant1 BTS a bien été créé.

Question 8 : Appliquer à l'utilisateur e1BTS le mot de passe btse1.

Question 9 : Vérifier que étudiant1 BTS peut ouvrir une session.

Question 10 : Créer l'utilisateur élève2 BTS, dont le *login* est e2BTS, qui appartient aux groupes btsgb et btsga, et dont le répertoire de base sera /home/btssnir/e2BTS .

Question 11: Appliquer à l'utilisateur e2BTS le mot de passe btse2.

Question 12 : Vérifier que etudiant2 BTS peut ouvrir une session.

Question 13 : Dans le fichier /etc/passwd, à quel groupe appartient l'utilisateur e2BTS ?

Question 14 : Comment savoir à quels autres groupes etudiant2 BTS appartient il ?

Question 15 : Modifier l'utilisateur e1BTS pour qu'il appartienne au groupe btsgb mais aussi au groupe btsga.

Question 16 : Vérifier que l'utilisateur e1BTS appartient au groupe btsga.

Question 17 : Quels sont les fichiers créés par défaut dans le répertoire personnel de e1BTS ?

Question 18 : Quels sont les fichiers créés par défaut dans le répertoire personnel de e2BTS ?

Question 19 : Quels sont les fichiers présents dans le répertoire /etc/skel ?

Question 20 : Créer un fichier vide nommé essais.txt dans le répertoire /etc/skel.

Question 21 : Créer un nouvel utilisateur nommé e3BTS.

Question 22 : Quels sont les fichiers créés par défaut dans le répertoire personnel de e3BTS ?

Question 23 : Quel est l'intérêt du répertoire /etc/skel ?

Question 24 : A quel groupe appartient l'utilisateur e3BTS.

Question 25 : Supprimer l'utilisateur e3BTS.

Question 26 : Supprimer les groupes et les utilisateurs créés pendant cette activité avec les commandes appropriées

Question 27 : Vérifier que les groupes et les utilisateurs ont été bien supprimés

5- Troisième application

Utilisateurs et groupes

Groupes	Utilisateurs	Répertoires
professeur		/home/professeur
	Prof1	/home/professeur/prof1
	Prof2	/home/professeur/prof2
	Prof3	/home/professeur/prof3
btsSN1		/home/btsSN1
	Etudiant1	/home/btssn1/etudiant1
	Etudiant2	/home/btssn1/etudiant2
btsSN2		/home/btsSN2
	etudiant3	/home/btsSN2/etudiant3
	Etudiant4	/home/btsSN2/etudiant4

Dossiers d'échanges

/home/professeur/partageprof
 /home/btsSN1/partagebtssn1
 /home/btsSN2/partagebtssn2
 /home/partagebts
 /home/partagetous

Droits sur les répertoires

- Tous les répertoires personnels ne sont accessibles qu'à leurs propriétaires respectifs.
- Les répertoires partageprof, partagebtssn1 et partagebtssn2 ne sont accessibles que pour les groupes auxquels ils appartiennent.
- Le répertoire partagebts n'est accessible qu'aux bts.
- Le répertoire partagetous est accessibles aux trois groupes uniquement. Seul prof1 a le droit d'écrire dans ce répertoire.